Edinburgh Napier
UNIVERSITY

HYPERTEXT MARKUP LANGUAGE

HTML

Web Tech
SET08101

Simon Wells
s.wells@napier.ac.uk
http://www.simonwells.org

TL/DR

- Web is about hypertext rather than (plain) text
- So need to turn text into hypertext
- We do this by tagging our text
- We use the hypertext markup language to tag our text

AIMS

- At the end of this (sub-section) of the topic you will:
    - understand how HTML has developed & why it works the way it does
    - be aware of the range of tags supported by HTML
    - be able to assemble basic HTML documents

HTML

- **H**yper**T**ext **M**arkup **L**anguage
- A **language** for turning **text** into **hypertext** using **markup**
- The standard markup language for creating web pages
- Not a programming language - no support for programming constructs
- Part of the triad of foundational web technologies (alongside CSS & Javascript)
    - Describes the semantic structure of the data, which CSS presents, and Javascript manipulates
- Browser receives HTML document from server or storage
- Document is then rendered visually (NB. Other user agents may use the returned HTML in other ways)

W3C

- Defines the standards
  - HTML 4.01
    - Based on Standard Generalised Markup Language (SGML)
  - HTML 5
    - Backwards compatible but no longer based on SGML
- & various related standards: XHTML 1.0, 1.0, 2 (cancelled)
  - Based on eXtensible Markup Language (XML)

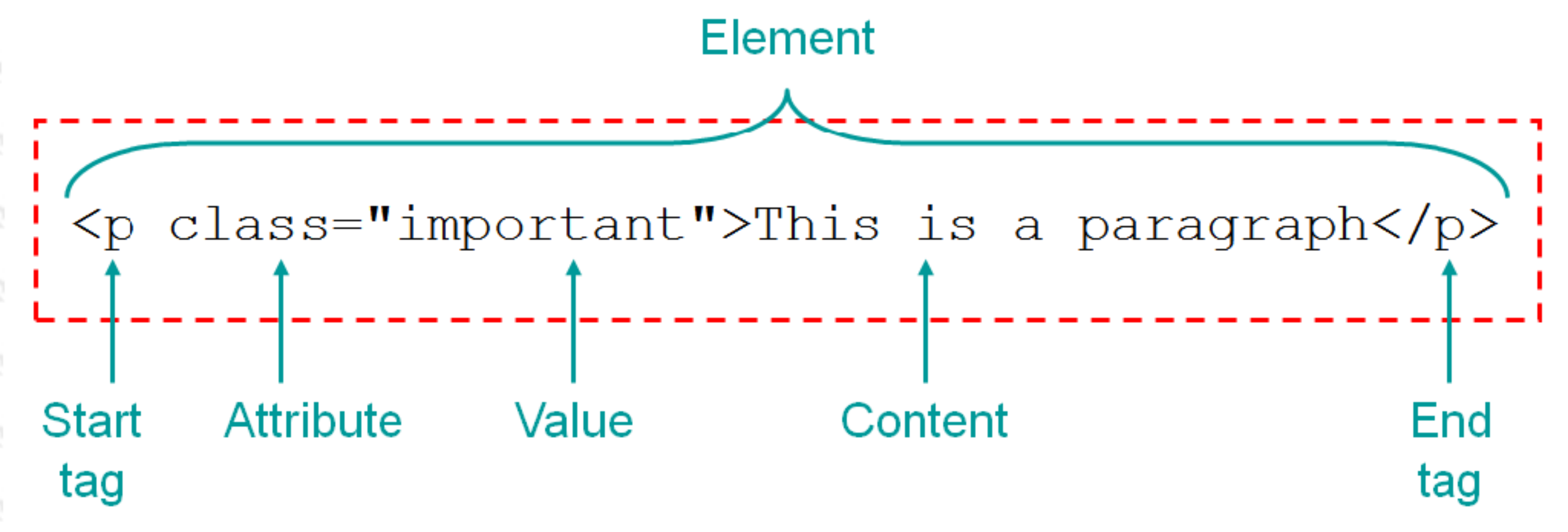
CLASSICAL TO MODERN

- Until 4.01 defined the visual presentation of a web page
  - Mixed structure & presentation
  - e.g. font face & font size, colour, size of elements, etc.
- Modern HTML
  - Describes the content, it's structure, and it's relation to other content
  - Visual presentation of those things is delegated to CSS

HTML ELEMENTS

- Elements are the building blocks of HTML documents
- Elements are **keywords** encapsulated within angle brackets, e.g. <html>
- Elements are represented using opening and closing tags, e.g. <html></html>
  - Most tags delineate the start & end of a portion of text or enclose other sets of tags so often paired - one for each end
  - Some stand alone amongst the text, e.g. <br />
  - All use angle brackets
- Combined to create structured documents by denoting structural semantics for the text (such as headings, paragraphs, lists, links, etc.)

HTML SYNTAX



Element

```
<p class="important">This is a paragraph</p>
```

Start tag

Attribute

Value

Content

End tag

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN”
     "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
   <HEAD>
      <TITLE>My first HTML 4.01  document</TITLE>
   </HEAD>
   <BODY>
      <P>Hello World from HTML 4.01</P>
   </BODY>
</HTML>
```

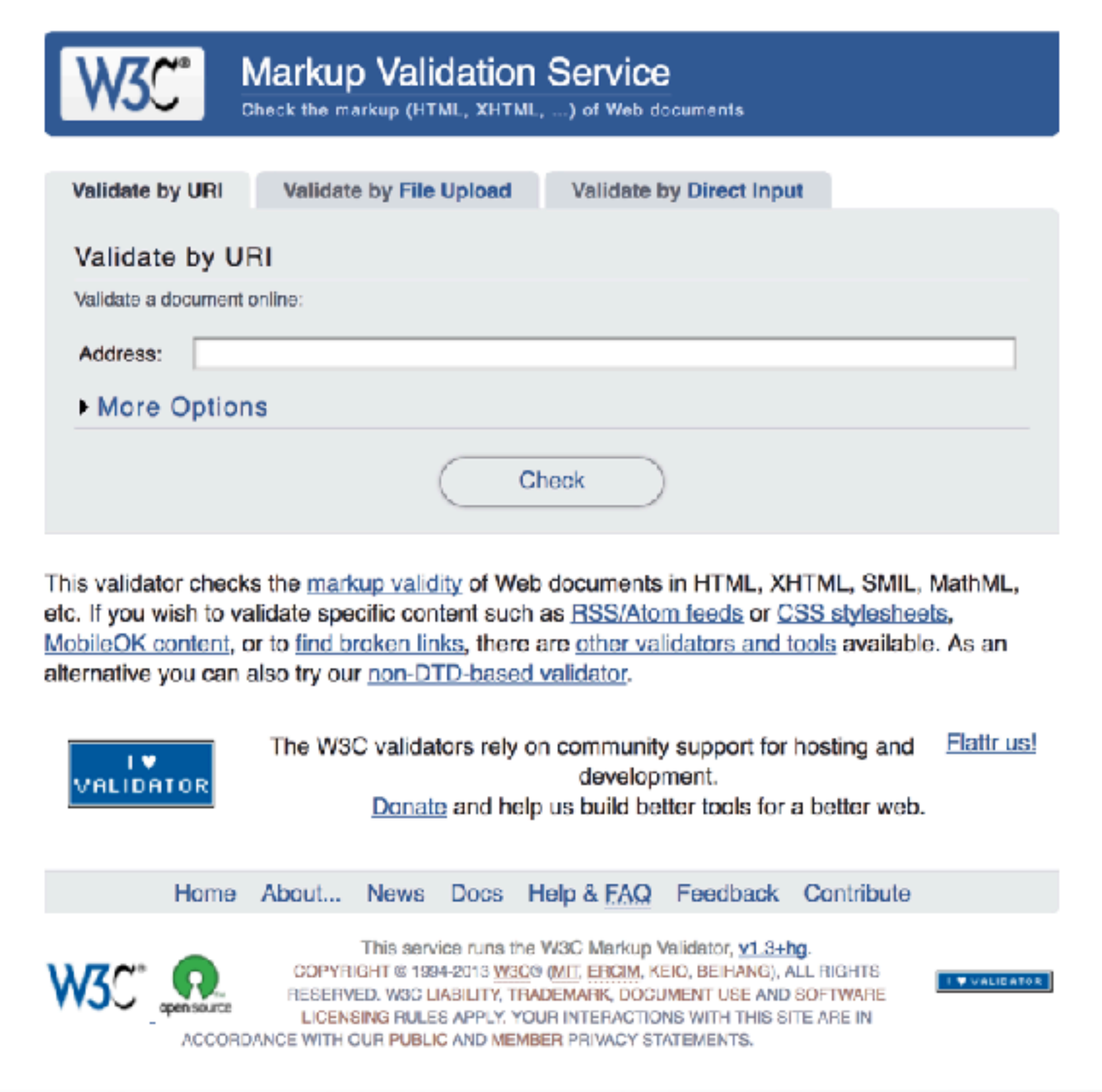
HTML 5

```
<!DOCTYPE html>
<html>
    <head>
        <title>My first HTML 5 document</title>
    </head>
    <body>
        <p>Hello World from HTML 5</p>
    </body>
</html>
```

Edinburgh Napier
UNIVERSITY

VALIDITY

- https://validator.w3.org/



W3C® **Markup Validation Service**
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI

Validate a document online:

Address:

This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, MobileOK content, or to find broken links, there are other validators and tools available. As an alternative you can also try our non-DTD-based validator.

The W3C validators rely on community support for hosting and development.
Donate and help us build better tools for a better web.

Flattr us!

This service runs the W3C Markup Validator, v1.3+hg.
COPYRIGHT © 1994-2013 W3C® (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

TAGS

- Document Structure: <html><head><body>
- Within <head> section: <title>, <base>, <meta>, <style>, <link>
- Tags for text blocks: <address>, <blockquote>, <div>, <h1>...<h6>, <p>, <pre>, <xmp>
- Tags that define lists: <dir>, <dl>, <dt>, <dd>, <menu>, <ol>, <ul>, <li>
- Tags that define text format: <b>, <basefont>, <big>, <cite>, <code>, <em>, <font>, <i>, <kbd>, <strike>, <sup>, <tt>, <u>, <var>
- Tags that define anchors and links: <a>
- Tags that define images and image maps: <img>, <area>, <map>

- Tags that define tables: <table>, <caption>, <thead>, <tbody>, <tfoot>, <tr>, <th>, <td>
- Tags that define forms: <form>, <fieldset>, <input>, <select>, <option>, <textarea>, <label>, <legend>, <isindex>
- Tags that define frames: <frame>, <frameset>, <iframe>
- Tags that define scripts: <script>, <noscript>
- Tags that define applets & plug-ins:<applet>, <param>, <object> (<embed> not standard)
- Tags that adjust text: <br>, <center>, <hr>

TEXT FORMATTING

- Headings: <h1>, …, <h6>
- Physical Styles: <b>, <i>
- Logical Styles: <cite>, <code>, <em>, <strong>

- <font face=“” size=“”>
  - You can do this but don’t. EVER.
  - Always use CSS for presentational aspects of typography
  - Support for presentational aspects within HTML is slowly being phased out

LISTS

- Definition Lists
    - <dl>, <dt>, <dd>
- Ordered Lists
    - <ol>, <li>
- Unordered lists
    - <ul>, <li>

LINKS

- I said last week that I consider this to be almost the most important element of HTML - *It turns text into hypertext*
  - **Internal Links**
    - Link: <a href=“#name”> … </a>
    - Target: <a name=“name”>…</a>
    - *Target also known as an anchor*
  - **External Links**
    - To another document in same site: <a href=“page.html”> </a>
    - To a target within another document: <a href=“page.html#name”> … </a>
    - To another site: <a href=“http://www.simonwells.org”> … </a>

TABLES

- For data representation (not presentation & layout)
  - Although admittedly there is some overlap
- Uses mix of <table>, <tr>, <th>, <td> tags
- Also: <thead>, <tbody>, <tfoot>, <caption> - more semantic structure

```
<table>
  <tr>
    <th>Heading 1</th>
    <th>Heading 2</th>
  </tr>
  <tr>
    <td>data 1</td>
    <td>data 2</td>
  </tr>
</table>
```

IMAGES

- <img> with mandatory attributes: src, alt
- Optional attributes::
  - width, height, longdesc
- Image types: GIF, JPG, PNG - browser support is so good that we don’t consider this so much anymore
- NB. Image size

FORMS

- Everything so far has been about retrieving HTML pages from the server - using the **HTTP GET** method

- Sometimes we want to send data from the client to the server (using the **HTTP POST** method)

  - More commonly using a **for**

  - We'll exploit this more when we start using Javascript

```
<form name="name"
  action="page.html"
  method="method">
  … various controls …
  </form>
```

FORM CONTROLS

- **Buttons:**

  <input type="submit">

  <input type="reset">

  <input type="button">

  <input type=“image">

- **Check boxes:**

  <input type=“checkbox”>

- **Radio buttons:**

  <input type=“radio”>

- **Text boxes**:

  <input type=“text”>

- **Password textboxes:**

  <input type=“password”>

- **Hidden fields:**

  <input type=“hidden”>

Edinburgh Napier UNIVERSITY

MORE FORM CONTROLS

- **File Upload:**

  <input type=“file”>

- **Selection Lists:**

  <select> <option> <optgroup>

- **Text Areas:**

  <textarea>

- **Label (for a control)**

  <label>

- **Group of controls:**

  <fieldset>, <legend>

RESOURCES

- Explore the Mozilla Developer Docs to find out more

SUMMARY

- You should now:
  - understand how HTML has developed & why it works the way it does
  - be aware of the range of tags supported by HTML
  - be able to assemble basic HTML documents
- There is obviously much more to effective HTML use than we can cover in one lecture, but we can develop effective skills through practise

NEXT

- Cascading Style Sheets (CSS) & their role in presenting the information that is captured in an HTML document